Audio summarization in real time for podcast, speeches and audio books

[1]Liya B.S, [2]Seenuvasan V, [3]Sathya Prakash K, , [4]Siva B

[1] Computer Science, Easwari Engineering College, [2] Computer Science, Easwari Engineering College,
[3] Computer Science, Easwari Engineering College,
[4] Computer Science, Easwari Engineering College

[1]liya.bs@eec.srmrmp.edu.in, [2]seenuvasanvelmurugan@gmail.com,
[3]sathyaprakash.vmr@gmail.com, [4]siva25022002@gmail.com

Abstract

As a result of the coronavirus disease (COVID-19) epidemic, mostgovernment work is now completed online. First-level medical checkups are regularly held online,online learning is now a widely available service atestablishments around the globe, and organisationsnow hold all of their monthly meetings online. Thereis an abundance of widely accessible online meetingsoftware on the market, including Google Meet,Microsoft Team, and others. This essay will examinea subject with numerous practical applications. Inthis paper, we present a method for creating a writtenor audio summary of a recorded video from its input.The described approach can also be used to createmeeting minutes, create subtitles, or extract lecturenotes for fun from lecture recordings. The systemgives users the option of using the text summary orcreating an audio output that matches the textsummary. The suggested approach is put into practisewith Python, and the suggested strategy is evaluatedwith the help of quick YouTube videos. Because there is no existing dataset for the relevant study andno benchmark measure for efficiency evaluation, the proposed method is manuallyevaluated on the downloaded video collection.

Index Terms: TF-IDF, deep learning, and speechrecognition in Python.

1. INTRODUCTION

Due to the development of current multimedia applications, interest in immersive communication technology has increased over the past 20 years. Due to the proliferation of mobile multimedia and ubiquitous computing, gaming, virtual and augmentedreality (VR and AR), teleconferencing, and entertainment applications have advanced significantly. The essential technologies for spatial sound collection, processing, and reproduction are provided by spatial audio research, which is at the forefront of recent advancements in 3D immersive user experiences.

Experts from several fields, including audio engineering, acoustics, computer science, applied that Voice recognition is a multidisciplinary field that use specific technology to recognise spoken words and convert it to text. The user's audio file is originally pre- processed for transcription. Using solely audio functions, extracting the text from the audio signal, controlling the summarization process with textual methods, and a hybrid strategy that combines the first two methods are the three ways that an audio summary can be directed. Every method has benefits and drawbacks. The advantage of not relying on transcription when creating a summary is that it is entirely independent from transcription; yet, since the summary is solely based on how things are said, this can also be problematic. Alternatively, the summary's heading textual approaches take advantage of the text'scontent and produce more informative summaries; however, transcripts may not There are several possible solutions to the problem of automatic summarization. These include simple unsupervised techniques, graph-based methods that utilise ranking to arrange the input text in a graph, or neural methods, which are constructed using graph traversal algorithms and are detailed in more detail in the paragraph that follows.

psychoacoustics, and others, collaborate on research in the interdisciplinary topic of spatial audio. Using the right mix of sound recording, processing,

and reproduction techniques, spatial audio aims to either synthesis a new auditory environment or recreate an existing one. In order to achieve this goal, it is crucial to preserve both the audio content's integrity and the spatial characteristics of the sound scene, which are brought about by the real placements of the sound sources and the characteristics of the acoustic environment.

Spatial audio methods and procedures may often be summed up in general terms. A comprehensive spatial audio pipeline typically consists of three stages: capture, processing, and reproduction. In theprocessing stage, the spatial information in the acquired sound scene is adjusted or supplied information that cannot be measured directly is taken from the sound scene.

Summarization is the process of condensing information into a concise, understandable form while preserving the syntax and semantics of the language. Italso involves communicating the key points of the content in a concise manner. The method of extractingtext from images or digitised documents is known as optical character recognition. The popular pdf2image python package is used to convert a pdf file into an image array when it is downloaded into our programme by a user. The command line programmes pdftoppm and pdftocairo for converting PDFs into PIL image lists are wrapped by the pdf2image module. The OCR must receive the list of the photos. Each image isdigitised to produce text that can be read. combining model recognition To recognise the letters andsubsequently the words in the image, characteristic detection is utilised. The recognised text is returned asa string.

Due to the ease of access to vast online resources

offered by Internet sites like YouTube, multimedia summary has grown to be a crucial demand. Automatic summary often seeks to produce a condensed and educational version of its source. In this article, we focus on audio summarising, a type of artificial summary where the source correlates to an audio signal. There are three different approaches to direct an audio summary: utilising solely audio functions, extracting the text from the audio signal, controlling the summarization process with textual methods, and a hybrid method that combines the first two methods.

Both benefits and drawbacks to each strategy. The ability to construct a summary only based on audio features without any reliance on transcription is abenefit, but since the summary is solely based on howthings are said, this can also be problematic. Contrarily, using textual approaches to lead the summary makes the most of the information in the textand results in summaries that are more helpful, however sometimes transcripts are not available. The quality of the summary can be improved by using textual and audio functions, although both ways have drawbacks.

Voice recognition is a multidisciplinary field that use specific technology to recognise spoken words and convert it to text. The user's audio file is originally pre- processed for transcription. Using solely audio functions, extracting the text from the audio signal, controlling the summarization process with textual methods, and a hybrid strategy that combines the first two methods are the three ways that an audio summary can be directed. Every method has benefits and drawbacks. The advantage of not relying on transcription when creating a summary is that it is entirely independent from transcription; yet, since the summary is solely based on how things are said, this can also be problematic. Alternatively, the summary's heading textual approaches take advantage of the text'scontent and produce more informative summaries; however, transcripts may not

always be available. Finally, using audio features and text-based methods can enhance the quality of the summary, although both ways have drawbacks.

I. LITERATURE REVIEW

There has been prior study on the classification of fishimages. In order to improve model accuracy, data augmentation was also applied in other application domains. The literature in these fields is succinctly outlined in this section.

[1] There are several possible solutions to the problem of automatic summarization. These include simple unsupervised techniques, graph-based methods thatutilise ranking to arrange the input text in a graph, or neural methods, which are constructed using graph traversal algorithms and are detailed in more detail in the paragraph that follows.

[2] The most common category of lengthy papers in thesummary literature is academic articles, thus we next look at related research on summarising these types oflengthy publications. Finally, we provide a summary of datasets for summarising academic publications.

[3] Despite the massive expansion of audiovisual data over the past 10 years, the number of tools and software applications created particularly to handle the issue of audio summarization is still quite small. Despite the fact that several studies have been done on various modules used to create audio summarizers, none have been found that directly deal with the development and improvement of audio summary systems.

[4] Graph-based models, which are frequently utilised in the field of extractive summarization, make extensive use of interactions between sentences and between queries and sentences. LexRank rates each sentence and plots the results in a similarity graph. In Wan and Xiao's use of manifold ranking, the sentence-to-sentence, sentence-to-document, and sentence-to- query links are utilised. We also model the aforementioned relationships as a graph at the token level, which is then combined into distributed representations of sentences, with the exception of the cross-document relationships.

[5] Interactions between sentences and between queries and sentences are heavily used in graph-based models, which are extensively used in the field of extractive summarization. Each sentence is rated by LexRank, and the results are displayed as a similarity graph. The sentence-to-sentence, sentence-to- document, and sentence-to-query linkages are used in Wan and Xiao's application of manifold ranking. With the exception of the cross-document relationships, we also model the aforementioned relationships as a graph at the token level, which is subsequently integrated into distributed representations of sentences.

[6] The need for more advanced audio file management arises from the growing consumption of audio/visual data. Long texts are condensed using the ground-breaking divide-and-conquer method. One such strategy aims to develop a technique for effectively condensing lengthy audio messages or samples in order to extract important information. It essentially comprises of three modules: Text Summarization, Speech-to-Text Conversion, and Text-to-Speech Conversion. With the exception of speech-to-text conversion, which requires the input of the audio file for which a summary must be produced, each module is supplied by the output of the one before it. The text summarization module's output is converted into an audio file in the end module. A suitable User Interface is made for the entire process using flask, and a web application is made to help.

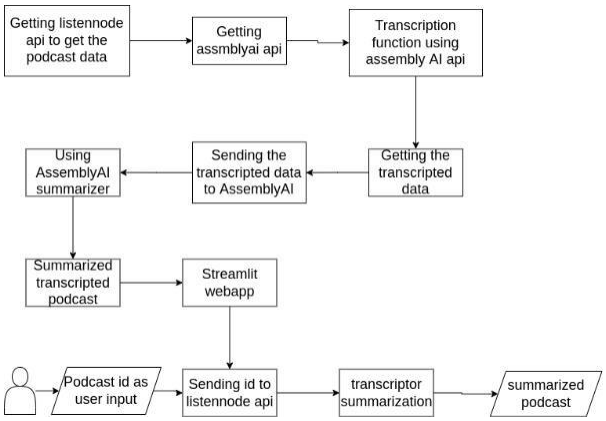
[7] The datasets available for query-focused summarising are currently too tiny to properly train data-driven summary algorithms. The manual creation of a query-focused summarization corpus, meanwhile, is a resource- and time-intensive process. We extract and summarise document sentences using a method proposed by colleagues. This makes it possible for the large model and the little benchmarks to match each other better. The model that was pre-trained on WikiRaf appears to have already attained a level of performance that is acceptable, according to the findings of experiments conducted on three DUC benchmarks. Strong comparison systems are outperformed by the model with data augmentation after optimization on specified benchmark datasets. Our suggested Q-BERT model and subnetwork fine-tuning also greatly improve the model's performance.

[8] Anyone who values their time and wants to learn more while spending less would benefit from having access to an automatic video summary, especially given how quickly the amount of available video content is expanding. The ability to appropriately view these films is becoming more and more important due to the exponential growth in the number of movies created by users. Video summaries are seen as a potentially useful way to maximise the information present in movies by finding and picking out instructive stills from the movie. It is demonstrated how key video elements can be extracted from the movie's subtitles and

used to create a summary of the content using text
summarization and video mapping algorithms.

[9] The development of the Internet and other social media platforms has given rise to a new "world of data" that consists of text, audio, and video files. It is challenging for a user to get an accurate overview or understand the relevant and important elements when using the accessible media. Additionally, users or reviewers of these data files are only interested in the summary or information that can be easily derived from the source files and is most relevant. The only method for summarising a single document or a collection of documents in order to extract important information from the source files is automatic text summarization, often known as ATS. The current ATS systems produce subpar summaries and require a lot ofprocessing time and space due to their erroneous encoding.

[10] The two fundamental modalities that make up video data are audio and vision. Multimodal learning, especially audiovisual learning, has drawn more attention in recent years. This is significant because multimodal learning has been shown to enhance the performance of a number of computer vision tasks. Contrarily, the majority of methods for summarising videos that are currently known depend only on visual information and ignore auditory information. In this condensed study, researchers contend that the audio modality might make it easier for the visual modality to understand the information and organisation of the movie, hence making summarization easier.



II. EXISTING SYSTEM

The current system can only recognise emotions whenthere is audio input. The current system relies on the Python Speech Recognition Library, which might not be compatible with all inputs due to dependency concerns. Because it hasn't been trained with a large dataset of audio files, the existing system cannot assess a huge dataset of audio files as an input.

III. PROPOSED SYSTEM AND ARCHITECTURE

The suggested approach examines the audio input, extracts the crucial keywords, and condenses them into a concise manner. The suggested method would deploy podcasts and audio books in real time. The suggested solution performs well in terms of accuracyand efficiency for any substantial audio inputs. We're going to utilise Streamlit to build a web application that will request the ID of the podcast the user wants to listen to before giving them a transcripted summary of the programme. The goal is to create an applicationusing Streamlit, a free open-source framework that makes it possible to create Python apps with few lines of code. In general, the suggested system is depicted in Figure 1's subsequent flowchart.

Fig. 1. Flowchart of the proposed system

A. Getting the PODCAST from the listen note api

We'll develop a technique that uses the user-provided podcast id to retrieve podcast data.
Take the episode's URL out of the listen notes
A comprehensive internet search tool and podcast

database is called Listen Notes. It makes it easy to build new services and applications based on it by providing an API for accessing podcast data. Through the usage of the Listen Notes API, we will extract the episode's url from our target podcast in this step.

- To utilise the Listen Notes API, we must first create an account in order to access the data. We must also join up for the free plan. You can process up to 300 requests per month with this subscription, which might be enough to complete small-scale personal tasks.
- The following action is to go to the podcast's Listen Notes page, click the episode you're interestedin, and then choose "Use API to retrieve this episode."

Then, in order to use the library later, we can changethe language code to Python and choose requests from a list of offered possibilities.

- By submitting a GET request to the Listen Notes Podcast API endpoint, a tactical action knownas information retrieval from the Listen Notes API iscarried out. When all is said and done, we save the outcome as a JSON object. The URL for the episodeis contained in this object, and it will be needed at a later time.

In addition, a JSON file called secrets. json is being imported. This file has a lot in common with a dictionary in that it too has a collection of key-value pairs. It must keep the corresponding API keys for AssemblyAI and ListenNotes in its possession. To access your accounts, youmust log in.

B. Transcription and summarization of thepodcast

Images will be preprocessed when the dataset has beenread. the next few steps

There will be a request for transcription in this part. We are no longer making GET requests to the AssemblyAI's transcript endpoint; instead, we are sending POST requests. Use of the post method is required when uploading the audio URL to AssemblyAI. This step is necessary to obtain the transcription and summary, both of which will be generated if we set the auto chapter to have a value of True, which is the default setting. Only if we set the auto chapter to False would we be able to retrieve the transcription. Following this, the id of the transcription response is kept.

Retrieve Transcription and Summary:

Finally, by making a GET request to Assembly AI, we can get the transcription and the summary. A few requests must be made before the response's status is finalised. The transcription is saved as a text file (.txt)and the summary is saved as a JSON file (.json).

Audio transcription: Audio transcription is the process of converting an audio recording into a text file. This could be an audio recording of anything, such as an academic paper, a music video, a conference, or an interview. In many different circumstances, having an audio recording rather than a written file is more practical. We'll utilise API.

Real-Time Streaming Transcription:

Streaming in real time Within a few hundred milliseconds, WebSocket API transmits text transcriptions back to clients. When there is an error,the API will always provide a JSON answer.

Summarization

The term "audio summary" refers to a condensed version of an audiobook. The most crucial concepts and themes from larger audiobooks are expertly condensed in these summaries into a format that is simpler to comprehend and absorb. They do a fantastic job of accurately reflecting the author's work in terms of its spirit, style, and

content. We willutilise the same AssemblyAI API to retrieve the data that has been summarised from the podcast transcription and communicate the data that has beentranscribed over.

C. Web App

We're going to utilise Streamlit to build a web application that will request the ID of the podcast theuser wants to listen to before giving them a transcripted summary of the programme. The goal isto create an application using Streamlit, a free open-source framework that makes it possible to create Python apps with few lines of code. All that is required to get started is to declare the functions that will repeat the steps that were just described and import the Python libraries. Additionally, we will offer a zipped package that includes the files comprising both the transcription and the summary.

Currently, we are capable of to outline the main parts of our project. In order to display the main title of the programme, we first utilise st.markdown. When we are done, we will use the st. sidebar method to build a left panel sidebar and then insert the episode id of our postcast into it. After entering the id, you must press the "Submit" button. When thebutton is pushed, the programme will go through all the steps required to summarise and transcribe the episode's audio. The results will be shown on the website following a short delay If you would want todownload the results, you can do so by clicking the Download button, which will allow you to combine the transcription and summary into a single file. The system automatically returns both the network URL and the local URL. All it takes to achieve the desiredeffect is to click on one of these links. Your favouritepodcast can now be summarised and transcribed for you by a terrific software!

VI. CONCLUSION

This project establishes the proof of concept and laysthe course for future development into a fully automated and taught system for podcast speech synthesis. We think there is much space for improvement given how complicated such a topic is. Unlike listening to music, podcasts typically demandprolonged periods of active focus from the listener. An audio file is used as the main input for processing for creating a summary, as detailed in this paper. Theaudio file is created by recording human speech thathas already been recorded or is now being uttered. Subjective characteristics like the speaker's delivery style, sense of humour, or production value may affect the listener's preference, but they are difficult to determine from a textual explanation The prepared summaries contain understandable audio data.

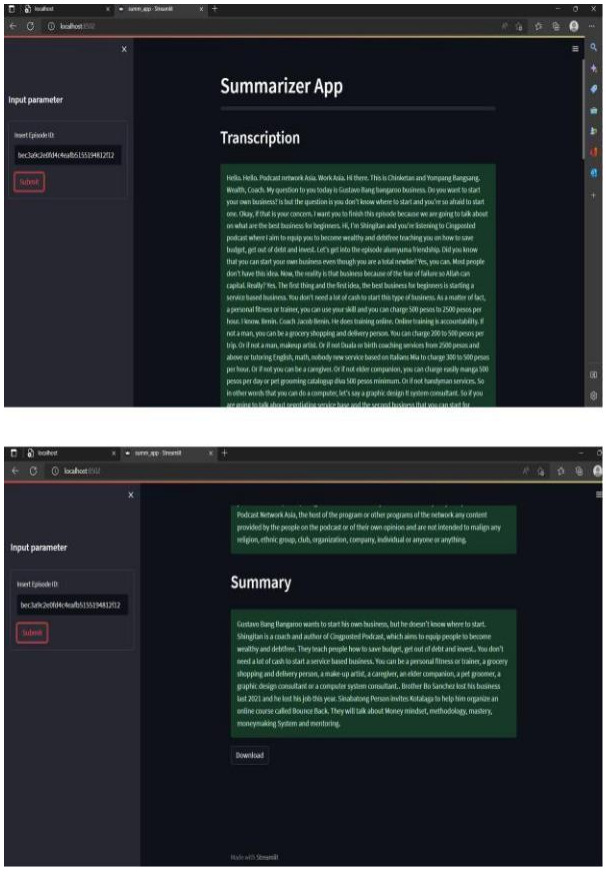


Fig. 2. Results

VII. RECOMMENDATION

There isn't a model out there that is 100% correct. Despite our predictions being 99.6% accurate, there were some errors. This model may not be able to identify species for which it has not been trained due to the small amount of data provided, but it will predict a type with which it almost matches. This model can yet be enhanced, though. This model will only be a system that summarises audio in real-time with greater data collecting and processing.

REFERENCES

[1] A. Vartakavi, A. Garg and Z. Rafii, "Audio Summarization for Podcasts," 2021 29th European Signal Processing Conference (EUSIPCO), 2021

[2] H. Lee and G. Lee, "Hierarchical Model For Long-Length Video Summarization With Adversarially Enhanced Audio/Visual Features," 2020 IEEE International Conference on Image Processing (ICIP), 2020

[3] M. -H. Su, C. -H. Wu and H. -T. Cheng, "A Two-Stage Transformer-Based Approach for Variable-Length Abstractive Summarization," in IEEE/ACM Transactions on Audio, Speech, and Language Processing,2020.

[4]Y. He, X. Xu, X. Liu, W. Ou and H. Lu, "Multimodal Transformer Networks with Latent Interaction for Audio-Visual Event Localization," 2021 IEEE International Conference on Multimedia and Expo (ICME), 2021

[5] A. Gidiotis and G. Tsoumakas, "A Divide-and- Conquer Approach to the Summarization of Long Documents," in IEEE/ACM Transactions on Audio, Speech, and Language Processing,2020

[6] R. K. Yadav, R. Bharti, R. Nagar and S. Kumar, "A Model For Recapitulating Audio Messages Using Machine Learning," 2020 International Conference for Emerging Technology (INCET), 2020

[7] H. Zhu, L. Dong, F. Wei, B. Qin and T. Liu, "Transforming Wikipedia Into Augmented Data for Query-Focused Summarization," in IEEE/ACM Transactions on Audio, Speech, and Language Processing,2022

[8] P. G. Shambharkar and R. Goel, "Analysis of Real Time Video Summarization using Subtitles," 2021 International Conference on Industrial Electronics Research and Applications (ICIERA), 2021

[9] P. Gupta, S. Nigam and R. Singh, "A Ranking based Language Model for Automatic Extractive Text Summarization," 2022 First International Conference on Artificial Intelligence Trends and Pattern Recognition.